

**UNITED STATES PATENT APPLICATION**

**FOR**

**ANTICIPATORY POWER CONTROL OF MEMORY**

**INVENTORS:**

**Nancy G. Woodbridge**

**Vasu J. Bibikar**

**INTEL CORPORATION**

**Prepared by:**

**John Travis**

**Reg. No. 43,203**

**EV325529171US**

# **ANTICIPATORY POWER CONTROL OF MEMORY**

## **BACKGROUND**

[0001] In a computer system with selective power control, individual memory banks that are not in current use may be placed in a reduced power state to save electrical power and extend battery life. Such implementations may wait until a need for a particular bank of memory arises before powering up that bank. Since transitioning from a reduced power state to a fully operational state takes a finite period of time, the system may have to wait for the needed bank of memory to be powered up and ready before accessing that bank of memory. This may force the processor or other accessing device to wait until the memory bank is ready, thus delaying the affected operations.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0002] The invention may be understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

Fig. 1 shows a flow diagram of a method of anticipating access to a memory bank, according to an embodiment of the invention.

Fig. 2 shows a flow diagram of a method of placing a memory bank into a low power state after a period of non-use, according to an embodiment of the invention.

Fig. 3 shows a diagram of a circuit, or combination of circuits, to selectively control power to multiple memory banks, according to an embodiment of the invention.

Fig. 4 shows a diagram of some of the inner details of an address comparator, according to an embodiment of the invention.

Fig. 5 shows a system comprising elements for anticipatory power control of memory banks, according to an embodiment of the invention.

Fig. 6 shows a flow chart of a method of determining how to select the threshold address whose access may trigger the operation of powering up a memory bank, according to an embodiment of the invention.

## DETAILED DESCRIPTION

[0003] In the following description, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure an understanding of this description.

[0004] References to “one embodiment”, “an embodiment”, “example embodiment”, “various embodiments”, etc., indicate that the embodiment(s) of the invention so described may include a particular feature, structure, or characteristic, but not every embodiment necessarily includes the particular feature, structure, or characteristic. Further, repeated use of the phrase “in one embodiment” does not necessarily refer to the same embodiment, although it may.

[0005] In the following description and claims, the terms “coupled” and “connected,” along with their derivatives, may be used. It should be understood that these terms are not intended as synonyms for each other. Rather, in particular embodiments, “connected” may be used to indicate that two or more elements are in direct physical or electrical contact with each other. “Coupled” may mean that two or more elements are in direct physical or electrical contact. However, “coupled” may also mean that two or more elements are not in direct contact with each other, but yet still co-operate or interact with each other.

[0006] An algorithm is here, and generally, considered to be a self-consistent sequence of acts or operations leading to a desired result. These include physical manipulations of physical quantities. Usually, though not necessarily, these quantities

take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers or the like. It should be understood, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

[0007] In a similar manner, the term “processor” may refer to any device or portion of a device that processes electronic data from registers and/or memory to transform that electronic data into other electronic data that may be stored in registers and/or memory. A “computing platform” may comprise one or more processors.

[008] As used herein, unless otherwise specified the use of the ordinal adjectives “first”, “second”, “third”, etc., to describe a common object, merely indicate that different instances of like objects are being referred to, and are not intended to imply that the objects so described must be in a given sequence, either temporally, spatially, in ranking, or in any other manner.

[0009] Some embodiments of the invention may be implemented in one or a combination of hardware, firmware, and software. Embodiments of the invention may also be implemented as instructions stored on a machine-readable medium, which may be read and executed by a computing platform to perform the operations described herein. A machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium may include read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g.,

carrier waves, infrared signals, digital signals, the interfaces that transmit and/or receive signals, etc.), and others.

[0010] Various embodiments of the invention may anticipate that a memory bank in a low power state is to be accessed, and power up the memory bank before such access is estimated to occur. The anticipation may be based on accessing another memory bank at a memory location that is within a predetermined range of the memory locations in the memory bank to be powered up. In some embodiments, including some of those described in more detail herein, a threshold address may be determined, and any access that is above the threshold address may be considered to be access within the predetermined range. Other embodiments, in which addresses are accessed in descending address order rather than ascending address order, may consider an access below the threshold address to be within the predetermined range. Although in some embodiments the addresses continue sequentially between the memory bank being accessed and the memory bank being power up, in other embodiments there may be a discontinuity in addresses between the two memory banks. The threshold address location may be at a particular place in a set of instructions, data, or both, that extends between the two memory banks. A memory bank may be considered to be a portion of memory that is able to be placed in a low power state or operational state independently from other portions of memory. An operational state may be a state in which instructions and/or data may be read from the memory bank and/or written into the memory bank. A low power state may be a state in which electrical power is consumed by the memory bank at a lesser rate than it is in the operational state. A low power state may be a state in which reading and writing operations to the memory bank cannot be reliably performed, but in which the contents of the memory bank are preserved, although various embodiments of the invention are not limited in this

respect. Alternately, a low power state may be a state in which the contents of the memory bank are not preserved (e.g., a power-off state), and must be restored when transitioning the memory bank from the low power state to the operational state, although various embodiments of the invention are not limited in this respect.

**[0011]** Fig. 1 shows a flow diagram of a method of anticipating access to a memory bank, according to an embodiment of the invention. In the flow diagram 10, at 110 a memory bank may be accessed at the current address. The access may be a read or write operation, and the current address in the memory bank may contain an instruction or data. At 120, it may be determined if the address is within a predetermined number of locations from the end of the memory bank. If not, the flow may return to 110 for the next memory access. If it is within the predetermined number of locations from the end of the memory bank, it may be assumed that further operations will soon cause memory accesses to pass beyond the end of the current memory bank and into the next memory bank, and the next memory bank may be checked at 130 to see if it is in a low power state. If the next memory bank is in a low power state, as determined at 130, it may be placed into an operational state at 140. If the next memory bank is already in an operational state, as determined at 130, operation 140 may be skipped, although in some embodiments the circuitry to place the memory bank into an operational state may be triggered regardless of the current state, but will have no effect if the memory is already in the operational state.

**[0012]** The binary decision block at 130 simplifies decision-making by assuming there are only two states, operational and low power, but in reality the low power state may consist of multiple sub-states, each with a different combination of power saving techniques and recovery times. By the same token, there may be multiple operational states with different levels of operability. Such complexities are not

discussed herein to avoid obscuring an understanding of the relevant embodiments.

The accesses described herein may be any of multiple types, such as but not limited to:

1) read and/or write accesses by a processor, 2) read and/or write access by a non-processor device, 3) read and/or write accesses caused by a direct memory access (DMA) operation, 4) combinations thereof, 5) etc.

[0013] Fig. 2 shows a flow diagram of a method of placing a memory bank back into a low power state after a period of non-use in the operational state, according to an embodiment of the invention. In flow diagram 20, at 210 a timer value may be determined. This determination may be made during operation of the system, or may have been made at a previous time, such as during code development. At 220 the timer value may be written to a storage element coupled to the timer, thereby making the timer programmable, the timer value to be used as a starting value whenever the timer is started or restarted. At 230 the memory bank may be placed in an operational state through any feasible means, such as but not limited to the operation of block 140 in Fig. 1. The timer may then be started at 240. Each time the memory bank is accessed, as determined at 250, the timer may be restarted with its original starting value that was written at 220. As long as the memory bank is not accessed, the timer may continue to run until it expires at 260, at which time the memory bank may be placed into a low power state. The timer may be implemented in any feasible manner, such as hardware, software, a combination, etc., and may operate in a linear or non-linear (irregular) timing fashion. In some embodiments, there may be multiple low-power states, and the described process may be used to enter a particular one of those low power states, or may be used to enter different ones of those low power states depending on various factors not discussed here. In some embodiments the low power state may be entered without waiting for the timer to time out, for reasons external to



the counter (e.g., the memory bank may be placed into a low power state by a power controller when another portion of the system is placed into a low power state).

[0014] Fig. 3 shows a diagram of a circuit, or combination of circuits, to selectively control power to multiple memory banks, according to an embodiment of the invention. In circuit(s) 30, memory banks 0-3, labeled 330-333 respectively, may be of any feasible type in which the individual memory banks have at least an operational state and a low power state, and that may be placed into those states independently of each other. The memory banks may be implemented with any of various technologies that have these characteristics, such as but not limited to static random access memory (SRAM), dynamic random access memory (DRAM), other volatile memory types, non-volatile memory types with operational and low power states, etc. In the illustrated example, it may be assumed that the addressable memory space extends linearly and contiguously from bank 0 through bank 3, although various embodiments of the invention may not be limited in this respect.

[0015] Requests for read and/or write memory access may be received by the memory controller 310 as indicated, and the memory controller 310 may control those accesses to all four memory banks. Such memory access requests may come from a processor, a non-processor device, an input-output (I/O) controller, a DMA controller, or any other feasible source. Memory power controller 320 may be used to individually control the power states for each memory bank. In the illustrated embodiment of Fig. 3, memory power controller 320 is shown coupled to an address comparator 311 that is within memory controller 310, and which provides inputs for memory power controller 320, but other embodiments may be configured differently.

[0016] Fig. 4 shows a diagram of some of the inner details of an address comparator, according to an embodiment of the invention. In address comparator 311,

a threshold address may be loaded into storage element 420, while the address that is currently being accessed may be placed at 410. Although 410 may be a storage element, it may also simply be a buffer to provide the current address to compare circuit 430, without providing intermediate storage for that address. If compare circuit 430 determines that the current address is greater than the threshold address (or alternately, determines that it is greater than or equal to the threshold address, depending on the specific implementation), then address comparator 311 may provide a signal to the memory power controller 320 indicating such, thereby triggering the sequence of events that cause the next memory bank to be placed in an operational mode.

[0017] Fig. 5 shows a system 50 comprising elements for anticipatory power control of memory banks, according to an embodiment of the invention. In some embodiments memory 550, memory controller 540, and power controller 520 may correspond to memory banks 330-333, memory controller 310, and memory power controller 320 of Fig. 3, respectively, but in other embodiments these elements may differ in various ways. Processor 510 may comprise a single processor or multiple processors. I/O circuitry 530 may provide an interface between processor 510, memory controller 540, power controller 520, and other system components such as peripherals. Memory requests may come to the memory controller 540 from processor 510, various ones of the peripherals, the I/O circuitry 530, and other components not specifically described. In some embodiments, power controller 520 may also provide power control for other elements in the system beside the memory banks.

[0018] Fig. 6 shows a flow chart of a method of determining how to select the threshold address whose access may trigger the operation of powering up a memory bank, according to an embodiment of the invention. The process of flow chart 60

might be followed, for example, after executable code has been produced, the memory address locations of the executable code have been determined or at least estimated, and it has been shown that related code will cause memory accesses to span the address space between two or more memory banks. At 610, it may be determined where within two memory banks the related code will be placed, and at what point in the code the transition from one memory bank to another will take place.

**[0019]** At 620 it may be determined how long it will take for a memory bank to power up from a low power state to an operational state, and an estimate of how many memory accesses it will take to occupy that amount of time. The number of memory accesses may then be converted into the number of memory locations whose access will take that amount of time. The clock speed of the memory may be a factor in this determination. If the access is expected to be linear, such as a DMA operation that accesses each location one time in sequence until after it crosses into the next memory bank, then the number of locations may be equal to the number of accesses. If the access is expected to be non-linear, such as accessing branch instructions that will skip over multiple locations, or accessing an iterative loop that will repeat multiple times before finally crossing over into the next memory bank, that complexity may be factored into a determination of how many separate locations will be accessed before crossing into the next memory bank. The desired number of locations may be set to approximate the previously determined power up time, may be set to a longer amount of time, or may be set to a shorter amount of time, depending on factors such as, but not limited to, the operational goals and the relative penalties for estimating too high or too low.

**[0020]** At 630, a threshold location (e.g., the lowest location whose access will trigger a power up of the next memory bank) may be determined based on the number

of locations derived in the previous paragraph. Various techniques may be used. For example, the number of locations may be subtracted from the highest-numbered address in the current memory bank to produce the threshold location. At 640, the threshold value just determined may be made available for subsequent use by the system. This may be accomplished in any of various ways, such as but not limited to: 1) placing the value into a hardware storage element for use by the comparator circuit, 2) including the value in initialization code that, when executed, will place the value into the hardware storage element for use by the comparator circuit, 3) including the value in operational code that, when executed, will place the value into the hardware storage element for use by the comparator, 4) dynamically determining the value during operation and writing the value into the hardware storage element for use by the comparator circuit. Alternately, if a software counter and/or timer is to be used, any of these techniques may place the value into a software-accessible location for subsequent use by the software counter and/or timer.

**[0021]** The foregoing description is intended to be illustrative and not limiting. Variations will occur to those of skill in the art. Those variations are intended to be included in the various embodiments of the invention, which are limited only by the spirit and scope of the appended claims.